



# Proyecto LORAPERTURA

@fergar73 – fernando.gm010@gmail.com



# INDICE DE CONTENIDO

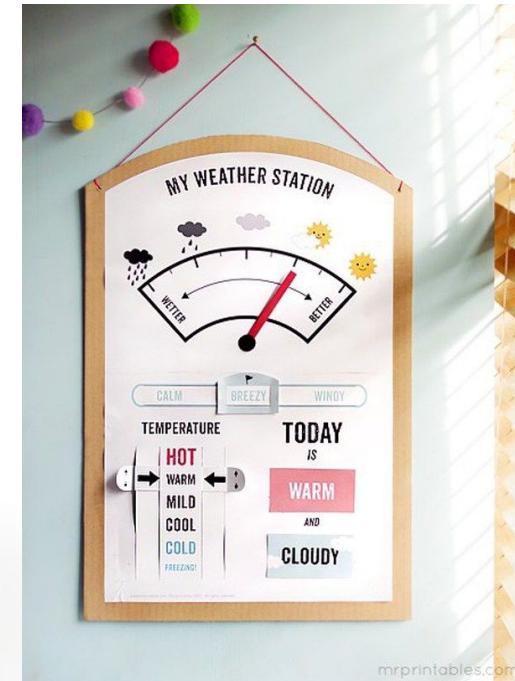
---

- 1 Introducción
- 2 Dispositivo Físico
- 3 Software lorapertura
- 4 Integración en The Things Network
- 5 Visualización y Control: lorapertura en myiotopentech

# INTRODUCCIÓN

---

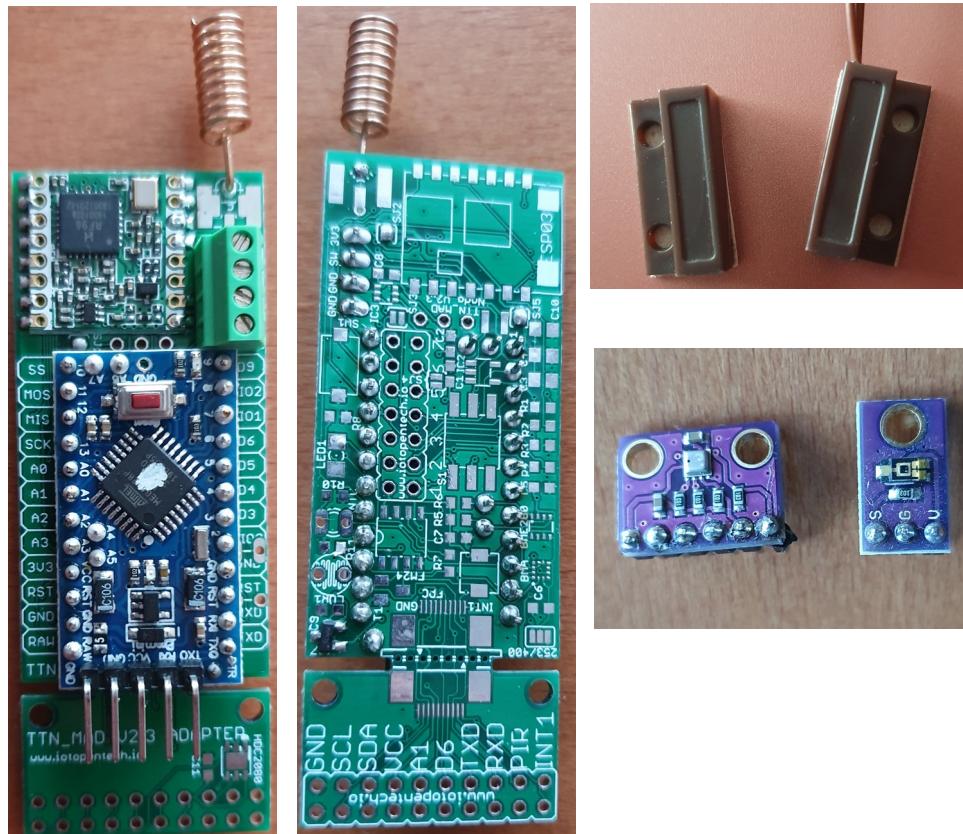
El dispositivo lorapertura permite saber cuando una puerta está abierta o cerrada e informar de su estado a The Things Network. Ofrece información de temperatura, presión y humedad del sitio donde esté instalado así como la luminosidad del espacio.



# DISPOSITIVO FÍSICO

## Listado de Componentes:

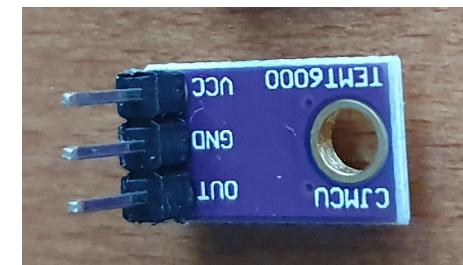
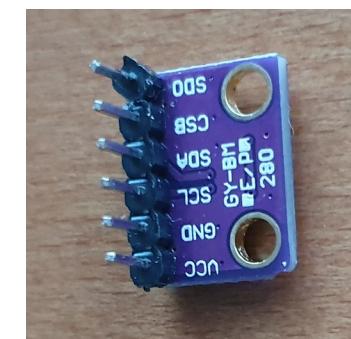
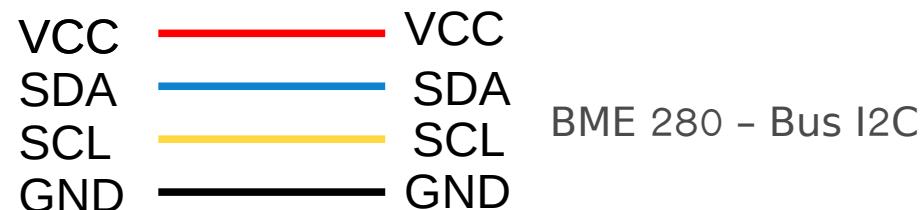
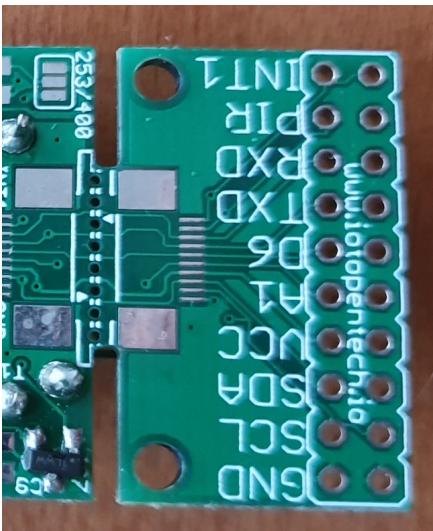
- 1 Nodo TTNMAD v2.3
  - 1.1 [Doc. Github Nodo TTNMAD v2](#)
  - 1.2 [Doc. Github Montaje Nodo TTNMAD v2 \(\\*\)](#)
- 2 Sensor magnético puerta abierta
- 3 Sensor BME280 – Temperatura  
Presión y Humedad
- 4 Sensor TEMT6000 – Luminosidad



(\*) El nodo ttnmad se ha montado sin el pulsador on/off que se indica en puntos 7 y 8 del montaje. En su lugar se ha utilizado el jumper SJ5 para controlar el encendido por interruptor separado del nodo.

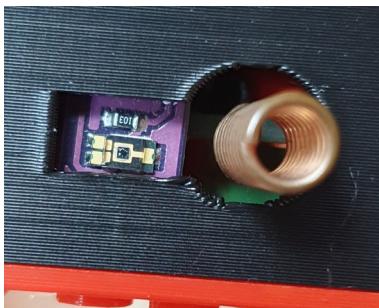
# DISPOSITIVO FÍSICO

## Conexiones:

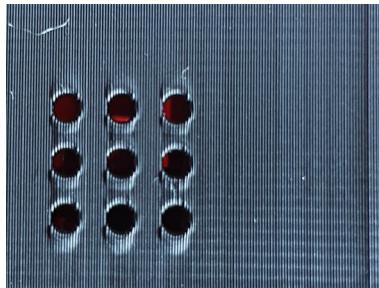


# DISPOSITIVO FÍSICO

Caja lorapertura:



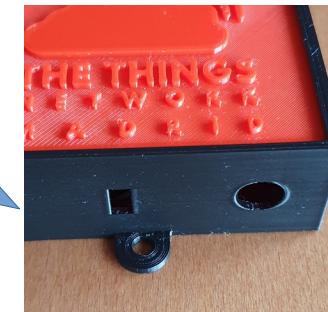
Ranura superior para el sensor de luminosidad y antena



Rejilla de ventilación en zona cercana a sensor BME280



Interruptor accesible en el exterior de la caja



Salida cables sensor magnético y entrada (opcional) para alimentación externa

## SOFTWARE LORAPERTURA

---

Código abierto disponible en github

Disponible activación por OTAA o ABP

Dependencias con las siguientes librerías

```
#include <lmic.h>          // Libreria para la implementación del protocolo Lorawan
#include <hal/hal.h>        // Hardware Abstraction Layer for LMIC
#include <SPI.h>           // Bus SPI - utilizado para la comunicación con el RFM95
#include <Wire.h>           // Bus I2C - utilizado para comunicación con el BME280
#include "LowPower.h"         // Librería para la gestión de bajo consumo del arduino
#include <forcedClimate.h>   // Librería para acceder al sensor BME280 con bajo consumo de memoria
#include <YetAnotherPcInt.h>  // Permite elegir el puerto para la gestión de la interrupción de puerta abierta
#include <CayenneLPP.h>       // Construcción del payload en formato Cayenne
```

# SOFTWARE LORAPERTURA

---

## Definición y envío del objeto Cayenne en el payload

```
// Variable en formato Cayenne para el envío del payload a TTN. Se indica tamaño de cada campo del payload.  
// LPP_ANALOG_INPUT_SIZE 4      -- 2 bytes  -- "analog_in_1"  
// LPP_DIGITAL_OUTPUT_SIZE 3    -- 1 byte   -- "digital_out_2" y "digital_out_3"  
// LPP_TEMPERATURE_SIZE 4       -- 2 bytes  -- "temperature_4"  
// LPP_BAROMETRIC_PRESSURE_SIZE 4 -- 2 bytes  -- "barometric_pressure_5"  
// LPP_RELATIVE_HUMIDITY_SIZE 3  -- 1 byte   -- "relative_humidity_6"  
// LPP_LUMINOSITY_SIZE 4        -- 2 bytes  -- "luminosity_7"  
CayenneLPP lpp(25);
```

```
lpp.reset();  
lpp.addAnalogInput(1, readVcc() / 1000.F);                      // Enviamos Voltaje actual  
climateSensor.takeForcedMeasurement();                            // Obtenemos los datos de temperatura, presión y humedad del BME280  
lpp.addTemperature(4, climateSensor.getTemperatureCelcius());    // Enviamos la medición de temperatura  
lpp.addBarometricPressure(5, climateSensor.getPressure());     // Enviamos la medición de presión  
lpp.addRelativeHumidity(6, climateSensor.getRelativeHumidity()); // Enviamos la medición de humedad  
lux = (((analogRead(LIGHTSENSОРPIN) * 3.3 / 1024) / 10000.0) * 1000000) * 2.0 ); // Obtenemos la medición de luminosidad de TEMT6000 y la pasamos a lux  
lpp.addLuminosity(7,lux);                                         // Enviamos la medición de luminosidad  
ev_anter_envio = eventoPuertaAbierta;                           // Guardamos el valor del eventoPuertaAbierta antes del envío  
lpp.addDigitalOutput(2, eventoPuertaAbierta);                   // Enviamos valor de eventoPuertaAbierta para saber si ha saltado la interrupción de Puerta Abierta  
lpp.addDigitalOutput(3, digitalRead(pinEventoPuertaAbierta));    // Estado de la puerta 1 => Cerrada || 0 => Abierta  
LMIC.pendTxConf = eventoPuertaAbierta;                          // Si se ha producido el evento de puerta abierta activamos el envío del mensaje con ACK  
LMIC_setTxData2(1, lpp.getBuffer(), lpp.getSize(), LMIC.pendTxConf); // Se hace el envío del mensaje a TTN
```

# SOFTWARE LORAPERTURA

Frecuencia de envío parametrizable en función de la política de uso de TTN

Air Time Calculator: <https://avbentem.github.io/airtime-calculator/ttn>

EU863-870 uplink and downlink							
	overhead size <sup>①</sup>		payload size <sup>①</sup>		share <sup>①</sup>		
	-	0	+	-	25	+	
data rate	DR6 <sup>①</sup> <b>SF7</b> <small>BW 250</small>	DR5 <b>SF7</b> <small>BW 125</small>	DR4 <b>SF8</b> <small>BW 125</small>	DR3 <b>SF9</b> <small>BW 125</small>	DR2 <b>SF10</b> <small>BW 125</small>	DR1 <sup>①</sup> <b>SF11</b> <small>BW 125</small>	DR0 <sup>①</sup> <b>SF12</b> <small>BW 125</small>
airtime	30.8 ms	61.7 ms	113.2 ms	205.8 ms	411.6 ms	823.3 ms	1,482.8 ms
1% max duty cycle	3.1 sec 1,167 msg/hour	6.2 sec 583 msg/hour	11.3 sec 318 msg/hour	20.6 sec 174 msg/hour	41.2 sec 87 msg/hour	82.3 sec 43 msg/hour	148.3 sec 24 msg/hour
fair access policy	88.8 sec (avg) <b>40.5</b> avg/hour 972 msg/24h	177.7 sec (avg) <b>20.3</b> avg/hour 486 msg/24h	325.9 sec (avg) <b>11.0</b> avg/hour 265 msg/24h	592.8 sec (avg) <b>6.1</b> avg/hour 145 msg/24h	1,185.5 sec (avg) <b>3.0</b> avg/hour 72 msg/24h	2,371.1 sec (avg) <b>1.5</b> avg/hour 36 msg/24h	4,270.3 sec (avg) <b>0.8</b> avg/hour 20 msg/24h

```
// Define el SpreadFactor fijo que se utilizará en la transmisión Lora
#define SpreadFactor DR_SF7
// Indicar el tiempo que tarda el mensaje en el aire según el Spread Factor indicado
// se puede utilizar para ello la utilidad: https://avbentem.github.io/airtime-calculator/ttn
const unsigned long airTimeMs = 61.7; // Expresado en milisegundos
```

# SOFTWARE LORAPERTURA

---

Envío de mensaje con confirmación cuando se produce un evento de Apertura de Puerta.

```
// Asociamos el pinEventoPuertaAbierta a la Interrupción.  
pinMode(pinEventoPuertaAbierta, INPUT_PULLUP);  
PcInt::attachInterrupt(pinEventoPuertaAbierta, abriendoPuerta, FALLING);  
Serial.print(F("Interrupcion preparada.."));  
  
// Función de la Interrupción de Puerta Abierta.  
void abriendoPuerta()  
{  
    eventoPuertaAbierta = true;      // Al saltar la interrupción de Puerta Abierta, se establece a cierto/true la variable eventoPuertaAbierta  
    Serial.println(F("Evento puerta"));  
}  
  
    ev_anteriores_envio = eventoPuertaAbierta;                      // Guardar  
    lpp.addDigitalOutput(2, eventoPuertaAbierta);                     // Enviamos  
    lpp.addDigitalOutput(3, digitalRead(pinEventoPuertaAbierta));    // Estado  
    LMIC.pendTxConf = eventoPuertaAbierta;                          // Si se han  
    LMIC_setTxData2(1, lpp.getBuffer(), lpp.getSize(), LMIC.pendTxConf); // Se  
    ev_despues_envio = eventoPuertaAbierta;                          // Guardar  
    Serial.print(F("Paquete enviado a TTN "));
```

# SOFTWARE LORAPERTURA

Bucle infinito estructurado en pasos sencillos:

- 1 Modo espera o dormir dependiendo del parámetro modoAhorroEnergia

```
void loop() {
    tiempoDormido = 0;
    Serial.print(contador);Serial.println(F(" Empezamos ciclo loop"));
    if (modoAhorroEnergia) {
        Serial.println(F(" Modo dormido"));
        do {
            LowPower.powerDown(SLEEP_8S, ADC_OFF, BOD_OFF);           // Modo arduino dormido por 8 segundos -- SLEEP_8S
            tiempoDormido = tiempoDormido + 8;
        } while ((tiempoDormido < frecuenciaEnvioMsg) & (not eventoPuertaAbierta));   // En caso de puerta abierta se avanza para el envio del mensaje
    }
    else {
        Serial.println(F(" Modo espera"));
        esperaConInterrupciones(long(frecuenciaEnvioMsg) * 1000);
    }
    Serial.print(contador);Serial.println(F(" Send to TTN"));
}
```

- 2 Envío del mensaje y su procesamiento hasta quedar completado

```
}
Serial.print(contador);Serial.println(F(" Send to TTN"));
do_send(&sendjob);      // Toma de medidas y envío del mensaje a TTN en el próximo ciclo
Serial.print(contador);Serial.println(F(" While not completado"));
do {
    os_runloop_once();    // Gestión del envío y recepción de mensajes enviados por la librería LMIC. Variable completado
} while (not completado);
completado = false;       // Actualizar completado a falso para el próximo ciclo de envío
ev antes envío = false;  // Se inicializa el control de envío de puerta abierta antes.
```

# INTEGRACION EN THE THINGS NETWORK

---

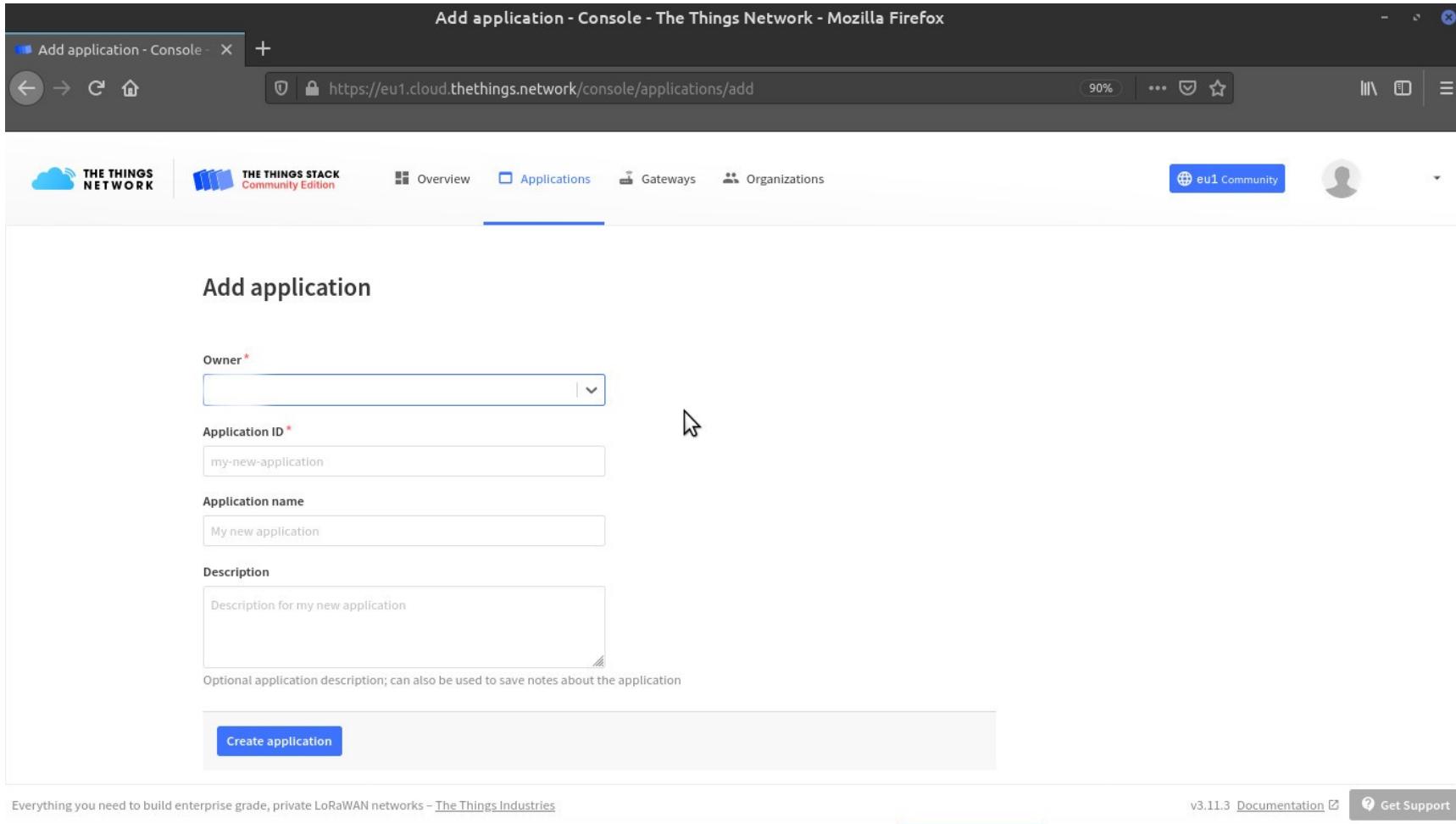
## Pasos a seguir

- 1 Registro en <https://www.thethingsnetwork.org/> (solo para nuevos usuarios)
- 2 Creación de la Aplicación en TTN
- 3 Alta del dispositivo asociado a la aplicación del punto 2
- 4 Integración http–webhook para envío de información a <https://my.iotopentech.io/>

Los puntos 2, 3 y 4 que se muestran en las siguientes slides están realizados en la versión 3 de the things network. Actualmente hay un proceso de migración de la v2 a la v3, para acceder a la v3 se ha utilizado la siguiente URL directa: <https://eu1.cloud.thethings.network>

# INTEGRACION EN THE THINGS NETWORK

## Crear aplicación



The screenshot shows a Mozilla Firefox browser window with the title "Add application - Console - The Things Network - Mozilla Firefox". The address bar displays the URL <https://eu1.cloud.thethings.network/console/applications/add>. The page content is the "Add application" form. At the top, there are navigation links for "THE THINGS NETWORK" and "THE THINGS STACK Community Edition", and tabs for "Overview", "Applications" (which is selected), "Gateways", and "Organizations". On the right, there's a "eu1 Community" button and a user profile icon. The main form fields are:

- Owner\***: A dropdown menu.
- Application ID \***: An input field containing "my-new-application".
- Application name**: An input field containing "My new application".
- Description**: A text area containing "Description for my new application". Below it, a note says "Optional application description; can also be used to save notes about the application".

At the bottom of the form is a blue "Create application" button. The footer of the page includes the text "Everything you need to build enterprise grade, private LoRaWAN networks - [The Things Industries](#)", the version "v3.11.3 Documentation", and a "Get Support" link.

# INTEGRACION EN THE THINGS NETWORK

## Alta nuevo dispositivo

The screenshot shows the 'Register end device' page in the The Things Network console. The URL in the browser is <https://eu1.cloud.thethings.network/console/applications/primera-prueba-stack-v3/devices/add/manual>. The page title is 'Register end device - Primera Prueba Stack v3 - The Things Network - Mozilla Firefox'. The left sidebar is open, showing 'Overview', 'End devices' (which is selected), 'Live data', 'Payload formatters', 'Integrations', 'Collaborators', 'API keys', and 'General settings'. The main content area is titled 'Register end device' and has tabs 'From The LoRaWAN Device Repository' and 'Manually' (selected). Under 'Preparation', there are sections for 'Activation mode' (radio buttons for 'Over the air activation (OTAA)', 'Activation by personalization (ABP)', 'Multicast', and 'Do not configure activation'), 'LoRaWAN version' (dropdown set to 'MAC V1.0.2'), 'Network Server address' (input field 'eu1.cloud.thethings.network'), 'Application Server address' (input field 'eu1.cloud.thethings.network'), 'External Join Server' (checkbox 'Enabled' is unchecked), and 'Join Server address' (input field 'eu1.cloud.thethings.network'). At the bottom is a 'Start' button.

# INTEGRACION EN THE THINGS NETWORK

## Nueva Integración con myiotopentech

Elegir la opción Custom webhook

The screenshot shows a Mozilla Firefox browser window with the URL <https://eu1.cloud.thethings.network/console/applications/primera-prueba-stack-v3/integrations/webhooks/add/template>. The page title is "Choose webhook template - Primera Prueba Stack v3 - The Things Network - Mozilla Firefox". On the left, there is a sidebar with a tree view of the application structure: "Primera Prueba Stack v3" (Overview, End devices, Live data, Payload formatters, Integrations (MQTT, Webhooks, Pub/Subs, Storage Integration, AWS IoT, LoRa Cloud), Collaborators, API keys). The "Webhooks" option is selected. The main content area is titled "Choose webhook template" and displays eight integration options in a grid:

- Ubidots: Integrate with Ubidots over UbiFunctions.
- Datacake: Send data to Datacake via TTI adapter. (The cursor is hovering over this card.)
- TagoIO: Integrate with TagoIO.
- Akenza: IoT made simple. Integrate with Akenza Core.
- ThingSpeak: Send data to ThingSpeak channel.
- Qubitro: Send your data to Qubitro.
- thethings.io: Integrate with thethings.io.
- Custom webhook: Create a custom webhook without template.

The bottom of the browser window shows the full URL again: <https://eu1.cloud.thethings.network/console/applications/primera-prueba-stack-v3/integrations/webhooks/add/template/datacake>.

# INTEGRACION EN THE THINGS NETWORK

## Nueva Integración con myiotopentech

The screenshot shows the 'Add webhook' configuration page in The Things Network. The left sidebar lists various integration options like MQTT, Webhooks, Pub/Subs, etc. The main form has sections for 'General settings' (Webhook ID: my-new-webhook, Webhook format: JSON) and 'Endpoint settings' (Base URL: https://example.com/webhooks, Downlink API key). A red box highlights the 'Downlink API key' field. Arrows point from the following text to the respective fields:

- Nombre para la integración → Webhook ID
- Tipo: JSON → Webhook format
- https://my.iotopentech.io/api/v1/"token-acceso-dispositivo"/telemetry → Base URL (with a red box around the "token-acceso-dispositivo" part)
- Generar nueva clave con la opción API Keys y copiarla aquí → Downlink API key
- Marcar a enabled todos los messages → Enabled messages

Copiar el token de acceso del dispositivo concreto mediante el botón disponible en myiotopentech

Details of the device page in myiotopentech. The 'VER CREDENCIALES' tab is active, displaying two buttons: 'COPIAR ID DEL DISPOSITIVO' and 'COPIAR TOKEN DE ACCESO'. A blue arrow points from the 'COPIAR TOKEN DE ACCESO' button to the 'token-acceso-dispositivo' part of the highlighted URL in the screenshot above.

## VISUALIZACION Y CONTROL: LORAPERTURA EN MYIOTOPENTECH

---

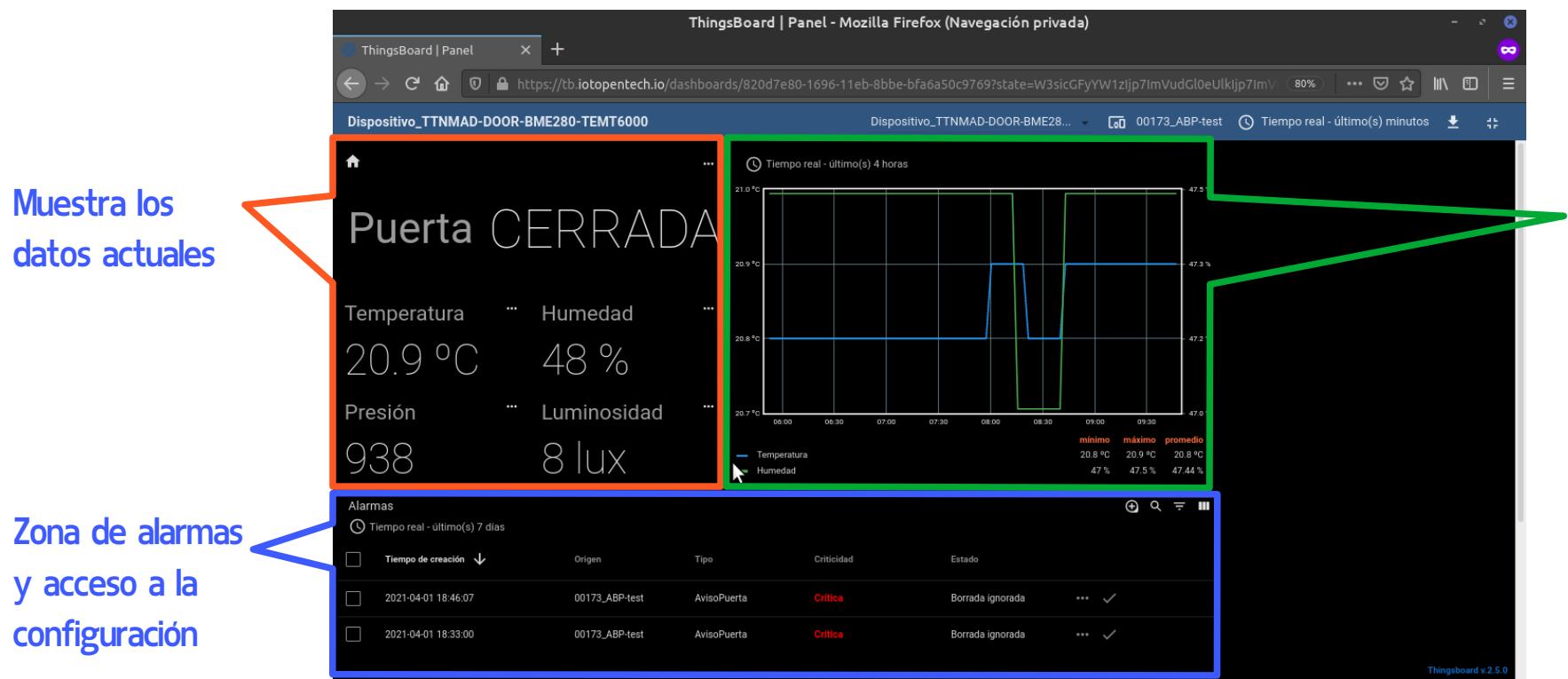
La capa de presentación y control del dispositivo lorapertura se ha realizado utilizando [myiotopentech - \(<https://my.iotopentech.io/>\)](https://my.iotopentech.io/)

El proyecto myiotopentech está basado en [thingsboard - \(<https://thingsboard.io/>\)](https://thingsboard.io/) que es una plataforma de código abierto que gestiona la recolección de datos de dispositivos de IoT, su procesamiento y visualización.

La información detallada de myiotopentech está disponible en el repositorio github <https://github.com/IoTOpenTech/myIoTOpenTech> donde entre otras cosas se muestran los dispositivos que están disponibles así como la información y pasos a seguir para crear nuevos tipos de dispositivos.

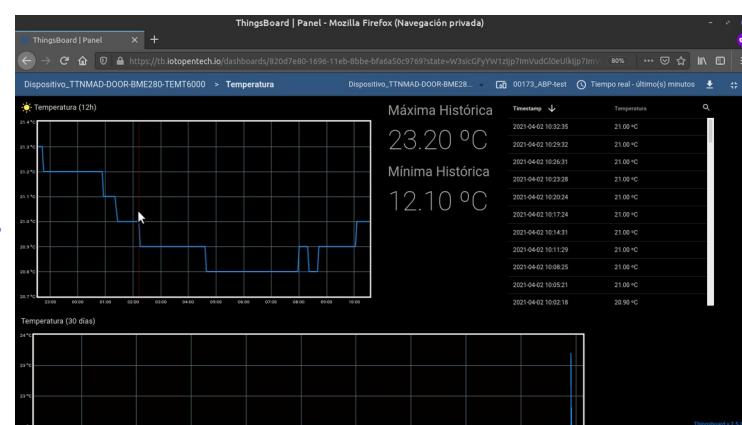
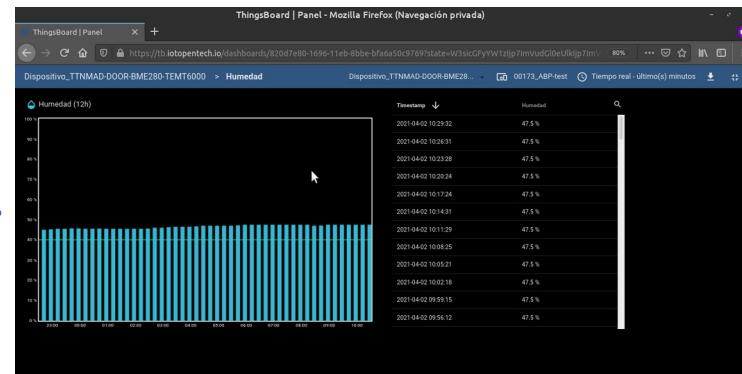
# VISUALIZACION Y CONTROL: LORAPERTURA EN MYIOTOPENTECH

El dispositivo lorapertura está disponible en myiotopentech. El nombre interno del dispositivo es TTNMAD-DOOR-BME280-TEMT6000.  
Dispone de un panel principal dividido en 3 zonas:



# VISUALIZACION Y CONTROL: LORAPERTURA EN MYIOTOPENTECH

La zona que muestra los datos actuales permite el acceso a información más detallada de cada una de las medidas a través de icono “...”



# VISUALIZACION Y CONTROL: LORAPERTURA EN MYIOTOPENTECH

En la zona de alarmas se muestra la información correspondiente a los distintos tipos de alarmas así como el acceso a la configuración.

	Origen	Tipo	Criticidad	Estado	...	✓	X
<input type="checkbox"/>	2021-04-04 18:10:39 00173_ABP-test	AvisoPuerta	Critica	Activa ignorada	...	✓	X
<input type="checkbox"/>	2021-04-04 17:55:26 00173_ABP-test	NivelBateria	Alerta	Activa ignorada	...	✓	X
<input type="checkbox"/>	2021-04-04 17:50:17 00173_ABP-test	Timeout	Alta	Borrada ignorada	...	✓	

3 Tipos de Alarmas definidos

AvisoPuerta: Para informar cuando se abre la puerta – Nivel Crítico

Timeout: Puede haber un problema porque no se reciben mensajes desde el dispositivo – Nivel Alto

NivelBateria: Informa cuando la batería está baja

Acceso al panel de configuración de las notificaciones

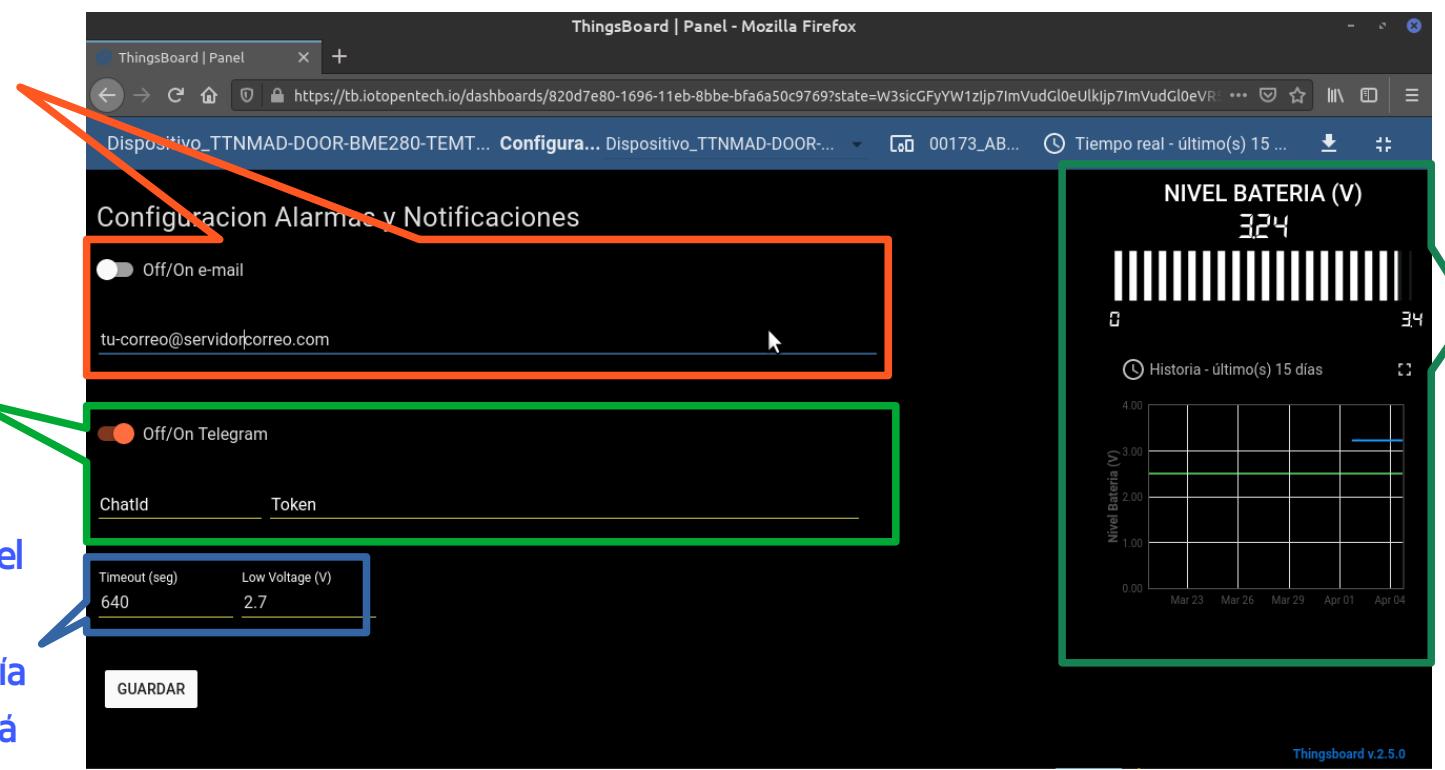
# VISUALIZACION Y CONTROL: LORAPERTURA EN MYIOTOPENTECH

En el panel de configuración se puede activar/desactivar las notificaciones de las alarmas. Se dispone de notificación por correo o telegram.

Indicar correo electrónico, si son varios separados por comas

Se requiere chatid y token de telegram. Opción off/on para activar o desactivar

Timeout para controlar que el dispositivo está enviando mensajes y umbral de batería por debajo del cual se saltará alarma de Nivel de Batería.



Información actual e histórica del nivel de batería



**MUCHAS GRACIAS!!!**

Podéis seguir las actividades de la comunidad  
TTN Madrid en el canal de slack `ttnmad` y en  
twitter `@ttn_mad`

